Research on Optimization of Multi-task Collaboration Platform Based on Instant Messaging Tools

Dapeng Zhou, Ran Ran, Yonghui Han and Libo Xu

State Grid Liaoning Electric Power Co., Ltd. Information Communication Branch, Shenyang, Liaoning

879527755@qq.com

Keywords: Instant messaging tools; Multitasking collaboration; Platform optimization

Abstract: As the number of users continues to increase, pressure on server will increase, and performance will decrease. Therefore, based on mixed mode of C/S and P2P of multi-tasking collaboration platform in instant messaging system, whose respective advantages are made full use in this paper to improve system performance, utilization of network bandwidth can be maximized to reduce server burden and improve user response speed so that user service experience will be enhanced.

1 Basic Architecture of Instant Messaging System

Instant messaging system uses system architecture that both C/S and B/S modes are adopted. Android and IOS mobile clients, as well as PC clients, use C/S mode. Besides it, system also provides Html5 client that supports B/S mode, and communication among clients supports C/S mode and P2P mode. In addition, clients cannot be connected directly, which need to be forwarded through server, namely C/S mode. If client can be connected directly, system will automatically use P2P communication to directly connect and communicate, where change in the form of communication is transparent to user[1].

1.1 Basic Model of Client-to-client Communication

As shown in Figure 1, Client 1 and Client 2 implement communication, which is the simplest model of instant messaging.
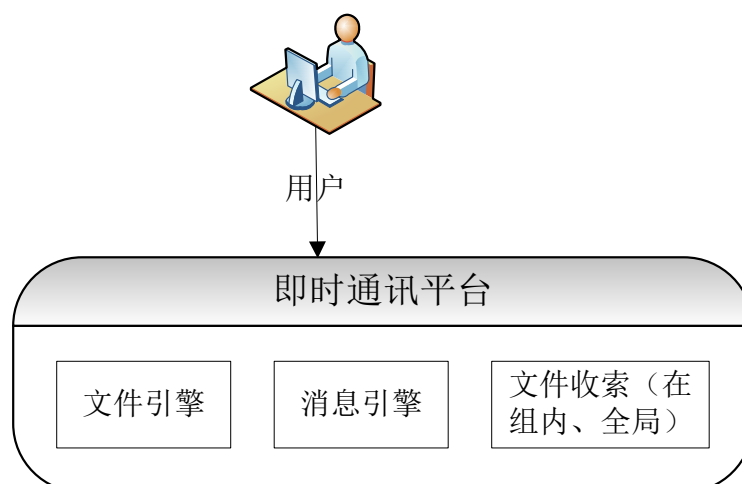

用户

即时通讯平台

文件引擎　消息引擎　文件收索（在组内、全局）

Figure 1 Client and client single server communication model

In this model, both Client 1 and Client 2 are connected to Server. Client 1 sends data to Server, and Server forwards the data to Client 2 to complete a data communication. Client 2 can also send data to Client 1 in this way.

A data transmission channel can also be created between Client 1 and Client 2. When Client 1 and Client 2 can be connected to each other, data will be transmitted directly between the two

without server forwarding, which can reduce not only burden on Server but also overall system cost. Therefore, early IM worked hard to achieve direct communication between Client 1 and Client 2, and even Server assisted in establishing connection [2-3]. However, with decline in hardware and bandwidth prices, and increase in server data value, IM systems have gradually reduced direct client connections and inclined to forward data by servers.

Besides forwarding data, server also needs to implement many auxiliary functions, such as account registration and verification, storage and delivery of friend relationships, user information, status modification, and query.

1.2 Client-to- client Communication Steps

According to basic communication model, steps required in user interaction process are analyzed. Following is detailed process of client-to-peer communication.

1.2.1. Client 1 enters his account and password to log in to server through instant messaging client, and server reads database registered by Client 1 to verify its identity. If user authentication passes, relevant information of Client 1, such as user's IP address and TCP/UDP port number will be registered and sign that Client 1 has successfully logged in will be returned as well. Meanwhile, status of Client 1 in system will be changed to online.

1.2.2. According to friend list stored on server by Client 1, server sends online information of Client 1 including IP address of Client 1 and TCP/UDP port number as well as online status to client 2 and Client D who are also online at the same time. Then Client 2 will displays the information to user after receiving this information.

1.2.3. Server sends friend list and related information stored on server by Client to Client1, where information such as friend's IP address, port number, and online status are included. Then Client 1 will obtain its friend list and online status.

1.2.4. If Client 1 wants to chat with his online friend Client 2, he can try to send chat message directly to Client 2. After receiving the message from Client 1, Client 2 will automatically reply that data has been received[4].

1.2.5. In some cases, Since Client 1 and Client 2 cannot communicate directly, Client 1 can only send chat data to Server, and server forwards it to Client 2.

2 Scalability Architecture Required by Mass Clients and Multiple Servers

The number of clients supported by single server is limited. Assuming that it can support 2,000 people online and communicating at the same time, when the number of clients exceeds capacity of single server, clients must be divided into multiple servers, such as Server 1 and Server 2.

However, multiple servers pose another problem. When Client 1 and Client 2 communicate, Client 1 sends data to Server 1 and hopes that Server 1 can forward it to Client 2. However, since Client 2 is connected to Server 2 and not connected to Server 1, Server 1 cannot find Client 2.

This problem requires adding another type of server so that Server 1 can discover Server 2. Original Server 1 and Server 2 are called Online Servers, and server used to contact Online Servers is called Information Server which also includes multiple servers and adopts distributed deployment to achieve flexible server configuration. Moreover, Root Information Server with only one server that is similar to Root server of DNS system is added to communicate with each other, and it needs to handle relatively small amount of communication.

Therefore, forwarding function of Server is implemented by several Online Servers and Information Servers. In addition, account function of Server is independent and implemented by Account Server. Moreover, relationship among friends is separated to implement on Information Server, while function of adding friends and friend confirmation is forwarded through Online Server and several Information Servers like data communication. Network communication model at this time is shown in Figure 2[5-6].
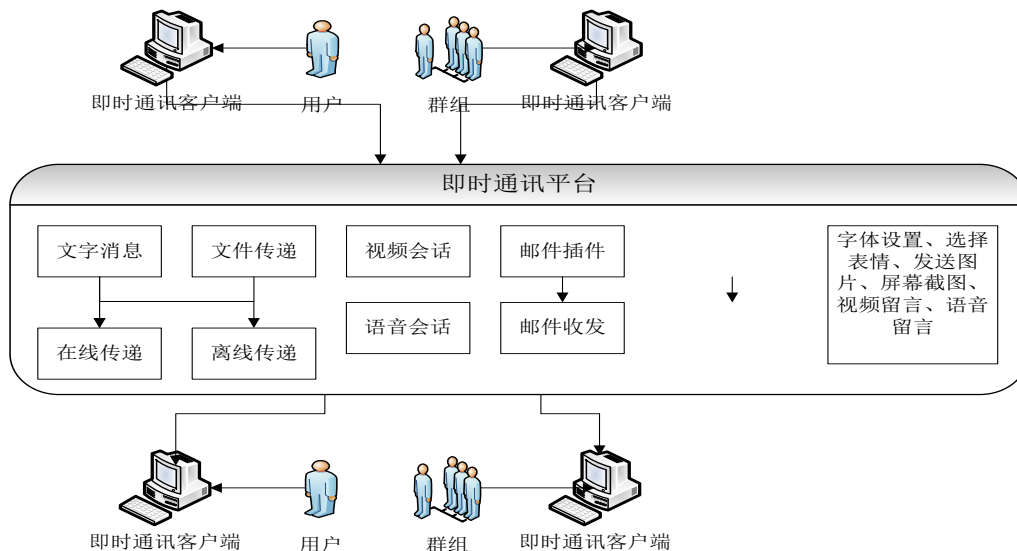
Figure 2 Network communication model for mass clients and multiple servers

The number of clients, Online Servers, and Information Servers are all flexibly configured and determined based on the size of clients, and online Server and client are one-to-many relationship. So are Information Server and Online Server. Additionally, both Account Server and Root Information Server currently have only one server.

3 Improvement and Enhancement on Instant Messaging System

IM system for personal use is called personal IM, and there are many cases of personal IM used for communication in reality. As mentioned in this paper, many problems occur in personal IM, and several servers are added to system, which are collectively referred as servers.

3.1 Address Book

Address book supports server to set all user accounts, user departments, user friend lists and other functions. Server that processes address book is called Contact Server[7], which is shown in Figure 3.
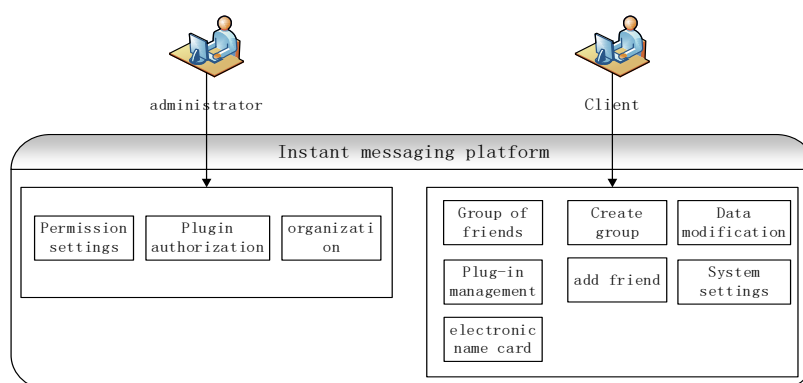

Figure 3 The server processing the address book

Address book is mainly for user authentication, user grouping, and friend search. Since user identity is set by server, identity authentication problem of users is naturally solved, and users are strictly restricted to internal. However, users belong to different departments, who can be grouped by department to carry out group chat within the department, and move offline department discussions online without being restricted by time and space.

Contacts which depends on situation can set up friend list for users. For example, friend list of members in development department is a member of this department, friends of members in sales department are members of sales department and marketing department, and friend list of

414

president's office includes office members and managers of each department.

3.2 Subsystem Inter-operability

The most common user requirement for subsystem communication is user account communication. For example, if human resources subsystem has been deployed, when implementing instant messaging system, it will be desirable to use user account of human resources subsystem which can be abbreviated as HR. Moreover, IM client first enters user account HR Account in HR, and then goes to HR Server to verify whether the account is correct. If it is correct, user account in IM will be returned. Client then uses the account in IM system to log in to Account Server and enters IM subsystem.

System architecture is mainly based on human resources subsystem. What's more, client and server of instant messaging system remain unchanged. When HR Client logs in, it first needs to go to server of instant messaging system so that a Token which will be used to log in to HR server with the help of user account. Besides it, after HR server receives Token, it needs to go to Account Server to verify validity of Token. Additionally, OAuth2 protocol used for WeChat login is similar in structure. Moreover, under premise of unified account numbers, instant messaging system and other subsystems can cooperate and call each other. For example, if human resources subsystem originally needs to send email notifications, it could be changed to send instant messages, which is more flexible and convenient [8-9].

3.3 Create Application on Instant Messaging System

Development trend of instant messaging systems is to create more applications based on system such as attendance, announcements, and approvals, and thus become basic platform for office automation. Since instant messaging system already has basic functions such as user management, network communication, and group permissions, creating applications on it can be said to do more with less. Specific method is that for each piece of data sent and received by instant messaging system, application attribute can be added to identify application module corresponding to piece of data. What's more, when user A uses approval application, input approval content to form approval data which will be sent from user A to user B just like ordinary text data. After receiving the approval data, user B determines application module corresponding to the data according to application attributes contained in the data, and then the data will be distributed to approval application.

This kind of architecture effectively ensures application scalability. Underlying data transfer function is only responsible for sending and receiving application data, which does not participate in creation and analysis of application data so that abstraction of underlying data transfer function can be ensured.

4 Features and Innovations of the System

During design and practice, a number of similar software systems are referenced in this system. Meanwhile, experience and opinions of many users are absorbed to ensure advantages of this system in all aspects. Moreover, development direction of instant messaging system is mainly to strengthen system collaboration and custom applications, especially to enhance system collaboration. For software in application, it is focus of future work to formulate corresponding interfaces and implement mutual calls.

Acknowledgement

References

[1] Jiang Junyan, Zong Dan, Lin Rui. Brief comparison and analysis of distributed file systems [J]. Internal combustion engine and accessories. 2018 (01)

[2] Liu Yong, Lu Xiaohui. File access processing and secure access method and device of distributed file system [J]. Guangdong Communication Technology. 2018 (03)

[3] Hui H.W., Zhou C.C., Xu S.G., Lin , A Novel Secure Data Transmission Scheme in Industrial Internet of Things, China Communications, vol. 17, no. 1, pp. 73-88, 2020.

[4] Lin F.H., Zhou Y.T., An X.S., Ilsun You, Kim-Kwang Raymond Choo, Fair Resource Allocation in an Intrusion-Detection System for Edge Computing: Ensuring the Security of Internet of Things Devices, in IEEE Consumer Electronics Magazine, vol. 7, no. 6, pp. 45-50, 2018. doi: 10.1109/MCE.2018.2851723.

[5] Su J.T., Lin F.H., Zhou X.W., Lv X., Steiner tree based optimal resource caching scheme in fog computing, China Communications, vol. 12, no.8, pp. 161-168, 2015

[6] Li Silong. Research and design of multi-protocol distributed police instant messaging system [D]. Guangxi Normal University 2018

[7] Chen Tian. Research and implementation of secure instant communication tools for enterprise users [D]. Xidian University 2018

[8] Sheng Shi. Design and implementation of P2P instant messaging software based on UDP protocol [D]. Beijing University of Technology 2018

[9] Wang Xiran. Design of reliable network transmission based on User Datagram Protocol (UDP) [J]. Computer Products and Distribution. 2018 (02)